DOCUMENT RESUME

ED 055 605 LI 003 159

AUTHOR Ershov, Andrei P.
TITLE On Software Compatibility.
SPONS AGENCY United Nations, New York, N.Y.
PUB DATE [69]
NOTE 8p.; (0 References); Background paper prepared for the United Nations meeting of the Ad Hoc Panel of Experts on Computer Technology held in February 1970

EDRS PRICE MF-$0.65 HC-$3.29
DESCRIPTORS *Computer Programs; *Computers; Computer Science; *Information Science; International Organizations; *International Programs; *Standards; Technology
IDENTIFIERS *Software Compatibility; United Nations

ABSTRACT

The problem of compatibility of software hampers the development of computer application. One solution lies in standardization of languages, terms, peripherals, operating systems and computer characteristics. (AB)

ED055605

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

"PERMISSION TO REPRODUCE THIS COPYRIGHTED MATERIAL HAS BEEN GRANTED BY
United Nations
TO ERIC AND ORGANIZATIONS OPERATING UNDER AGREEMENTS WITH THE U.S. OFFICE OF EDUCATION. FURTHER REPRODUCTION OUTSIDE THE ERIC SYSTEM REQUIRES PERMISSION OF THE COPYRIGHT OWNER."

On Software Compatibility

Dr. Andrei P. Ershov

Computerization of our life is one of defining features of the present. It is a typical characteristics of developed countries but is of a great importance for developing countries too. Except for socialist countries this computerization takes place mainly by an expansion of the capital of giant international computer manufacturers (IBM, GE, CDC etc.). This expansion is very intensive and it provides a high rate of the introduction of computers into anything but the profit-oriented optimization criterium which is applied here is sometimes rather far from real interests of the society in a given nation. Because of it this "natural" form of computerization has to be supplemented with a series of co-ordinated national and international programs which should be based on other than purely capitalistic principles. The UN might play a considerable role in the initiation and sponsoring such programs.

In my report, I shall touch only one but very important, in my opinion, side of the general problem. That is the problem of commonvalueness and compatibility of software. As a car wherever made can be driven and supplied by gas in any country as a program wherever written has to be able to be run at any computer at any plase for a solution of a similar problem. Now we are very far from it but definite technical precondition to it are already appearing and demand for it is practically inexhaustive because software production is those bottle-neck that mostly hampers the development of computer application.

A key word to the solution of this problem is standartization - but namely

- languages standardization
- terms and notions standardization
- periferals standardization
- operating systems standardization
- computer characteristics standardization

I believe that the next five years will be a critical point in the solution of standardization problems in computing. From

LI 003 159

one side, the development of aids to computations and data processing reached some level of maturity that provides a stability required for the standardization. From the other side the gigantic volume of the labor consumption for software strongly insits to think everybody how to reduce it.

Languages standardization is the most direct aid to the software compatibility. It is necessary to ditinguish levels of standardization: i.e. problem- oriented languages level, machine-oriented or machine languages level, and so called internal languages level. Each of them has its own features and aims. The standardization of problem-oriented languages is a decisive factor for the compatibility of the most numerous category of programs, namely, users programs. This kind of the standardization is quite possible in principle: it requires, first, a technique of a precise specification of syntax and semantics of a language and then a unification of quantitative restrictions imposed by particular compilers. However full standardization of this kind depends on the standardization of operating systems and computer characteristics (see below).

The standardization of machine language is,in a definitecontext, most desirable because it allows to solve nearly all the problems of compatibility. At the same time as an absolute approach it could considerably hamper the technical progress. Now, however, a de-facto standard is appearing that is one based on the IBM/360. This de-facto standard will cover more than a half of the world computers. This situation will take place 8-10 years and it should be maximally used to economize the resources for the software production. However, a standardization of operating systems is vitally necessary for a materialization of this kind of standard-ization.

Not so much and definite could be said about standardization of internal languages. It could be especially valuable when constructing programming systems. The idea is to introduce an intermediate, internal language between problem and machine languages. Being oriented more to a description of internal

algorithmic and structural characteristics of a problem than to a satisfaction of more specific and variable requirements from external languages an internal language could become more stable than any problem or machine language. At the same time, an introduction of an internal language into a technology of the software design can greatly reduce labor consumption when developing programming systems. There exist already some interesting experiments in using internal languages but in a whole this direction still requires further researches.

Terms and notions standardization is the most urgent problem which also maximally suits to the spirit of the international collaboration. Without it it is impossible to establish an integral picture of the computer world. Unfortunatelly, the initial situation demonstrates a confusion in many terms and notions (especially with respect to operating systems). Big manufacturers which take care about a uniform style of their technical documentation establish their own terminological systems but they are disjoint, corporatively closed and sometimes incomplete. The traditional style of the activity of terminological commissions in standardization commettees/s/ is inappropriate for computer matters because in this case a higher speed and more active position in introducing the standardized terminology are necessary. At the same time there exist now some projects which could be an appropriate basis for further standardization, e.g. "ICC-IFIP Dictionary of Computer Terms".

I believe that a preparation and publication under UN or UNESCO sponsorship of a multilanguage (English, Russian, French, Spanish, German) glossary of computer terms could be a very important action. The mentioned above ICC-IFIP Dictionary could be a basis for such glossary. Having been properly supplied and supported by national professional societies (there is no doubts in such a support) this project of several thouzand terms size could be inplemented in three or four years. It is necessary to note, however, that a successful completion of the work with a product of high quality is possible only by establishing a real international project with a proper organization, aiming and support.

Periferals standardization is also a key factor in solving the problem of software compatibility. This factor is even more important than machine languages and computer characteristics standardizations. A difference between computers can be hidden using problem-oriented languages but periferals operations usually considerably take into account specific features of periferal devices. At present a control of data exchange is usually performed in a program by commands to the operating system. Consequently, if the problem of the standardization of command languages of operating systems is solved (see below) then the problem of periferals standardization can be partially solved by introducing a logical periferals concept into the language. This way seems to be attractive because it, in some extent, follows the line of the development from machine languages to procedural ones. The practice, however, shows that the introducing of logical devices and standard formats is necessary but not sufficient. As a rule, logical device concept standardized an exchange operation proper but leaves out of the board such often important features as synchronization of a processor and a periferal, errors elaboration, restoration of a spoiled information and so on. Providing a uniform control of such features necessarily requires a standardization of the equipment. Another, no less important reason for the periferals standardization is providing a full compatability of information carriers (paper tapes, cards, magnetic tapes, disks, films etc.) that allows to store and transport the information in a form suitable for a direct input into a computer.

It is paid very much attention to the problem of the periferals standardization and now there exist several national standards. There are also some international de-facto standards (mainly on the base of the IBM equipment) which could be a base for international standards. I believe that any international organization which could deal with the problem of the periferal standardization should maximally support this work.

Operating system standardization seems to be a most difficult and critical point in the consideration of the problem. Differences in operating systems can easily make uncompatible programs of a solution

of same problems written in same languages for same computers. For example the series of computers with a common architecture IBM/360, Spectra, System 4, Siemens 4000 are actually incompatible because of differences is the operating systems. Even inside one system IBM/360 the differences between DOS/360 and OS/360 in many cases require reprogramming.

Standardization of an operating system has many aspects and tightly connected with other standardization problems. If the structure of the computer is fixed, periferals caracteristics are defined, a class of the system (batch-processing, time-sharing etc.) is given, problem-oriented and other procedural languages are defined then from now all specific features of the system are defined by its command language disposingits functions as an organizer of the computational process, controller of the separate units, and a connector with an operator (or user).

Researches show that actions of any operating system can be described by means of a visible number of fundamental concepts. These concepts are of a universal character.

Then, an analysis of specific operating systems and of conditions of program runs at them shows that there is a visible number of typical events stimulating one or another action of the operating system. The following events are, for example, among them:

- Initiation of a parficular job
- Choice of a next job
- Interruption of a job
- Resuming a job
- Performing an exchange etc.

An existence of typical events and of universal concepts gives us a hope that having had standards for the interface and computers classification we can define a universal command language for an operating system of either class. The syntax of the universal language should be able to be adapted to the syntax of the algorithmic language which is used to write the programs using the commands of the operating system.

At present the development of a universal command language for operating systems is in a very initial stage. Nobody proposed a complete draft such a language. There is only a ten works devote to a description some fundamental concepts and typical events. Thus, the operating systems standardization is a subject for futur developments. Nevertheless I deeply concure with the opinion that the definition of a universal language for operating systems is possible and its development and usage will deeply influence on the software production technology to the common weal.

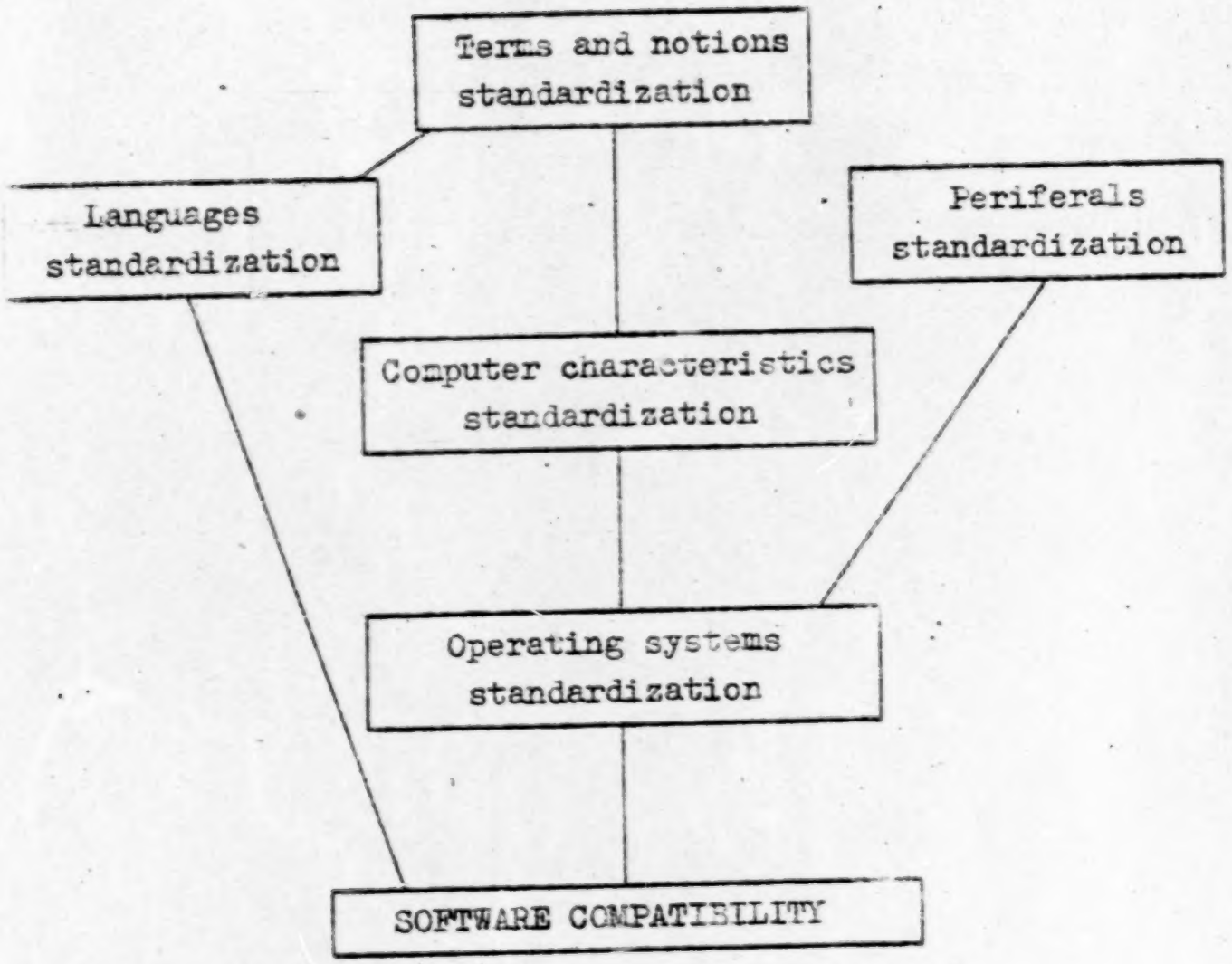
Computer characteristics standardization. Every specific program whatever programming language written in always takes into account an environment in which the program is destined to run. The environment includes in particular such characteristics of a computer as, for example, memory capacity, number of tapes used, mode of the connection with the external media and so on. There exist views that universal programs have to be written in a form which is an invariant to the environment so that they define their specific mode of the execution by enquiries to the environment. To my opinion, an orientation of a program to some specific diapason of the environment to which the program is best adapted is more realistic. Such approach requires, however, an introduction of a quantification of computer characteristic with their subsequent standardization. In particular, such approach can establish standards for the class or type of a computer. Then, having solved other standardization problems, a program specially tailored for a computer of some class can be run smoothly at any computer of the same class without loosing efficiency.

At present, computer characteristics publishers wishing to develop a uniform style of computers description made a lot of work on the way to the quantification of computer characteristics but this work has to be but again supplemended by definite efforts to standardization.

National standards. The materialization of the software compatibility can not ignore some specific regional or national requirements to computer application. A specific feature of descriptive capabilities of algorithmic languages is a possibility to use mnemonics. Naturally, mnemonics can be effective only

when using native languages and alphabets. Thus, a very important problem arises to establish national versions of universal programming languages and of other software. I believe that this problem can well be solved in accordance with standardization and compatibility requirements. In particular this problem stands in the USSR where English which is a father language for many programming languages is not so widespread as in other Western countries. A similar approach is applied in joint software projects of collaborating socialist countries where the necessity of establishing national versions is taken into account from the very beginning. It is necessary only to note that the consideration of national versions requires a special approach to the software technology when all its mnemonics is concentrated in distinquishable and changeble blocks but no disseminated over all the program.

Conclusion. Software is a critical path on the whole way of the development of the computer technology. Software compatibility can become a decisive factor in unloading this critical path. Standardization is a key word in the solution of compatibility problems. Below a flow-chart of connections between various standardization aspect considered in the report is presented.



A. Ershov

Dr. Andrei P. Ershov
Head, Programming Department
Computing Center
Professor,
Novosibirsk University